

REMARKS

Claims 1, 3-48, 50, 52, 70-73, 75-87, 89, and 91-108 were presented for examination. The Examiner rejected claims 1, 3-48, 50, 52, 70-73, 75-76, 79-87, 89, and 91-108 under 35 U.S.C. §103(a) as unpatentable over Lewallen (United States Patent No. 6,854,123, hereafter “Lewallen”) in view of Song et al. (United States Publication 2005/0066037, hereafter “Song”). The Examiner rejected claims 77-78 under 35 U.S.C. §103(a) as unpatentable over Lewallen, in view of Song, and further in view of Snyder (United States Patent No. 6,707,475, hereafter “Snyder”).

Applicants and their attorneys thank the Examiner for the opportunity to discuss this application and the cited references in the telephonic interview conducted on June 23, 2009. Applicants have amended the claims responsive to the Examiner’s suggestions during the interview. Claims 1, 3-5, 13, 15, 21, 23, 36, 38, 81-83, and 91 have been amended. No new subject matter was added. No claims were added or canceled. Claims 1, 21, 38, and 91 are independent. Applicants respectfully request reconsideration of all pending claims in view of the remarks below.

Rejection of Claims 1, 3-48, 50, 52, 70-73, 75-76, 79-87, 89, and 91-108 Under 35 U.S.C. §103(a)

The Examiner rejected claims 1, 3-48, 50, 52, 70-73, 75-76, 79-87, 89, and 91-108 under 35 U.S.C. §103(a) as unpatentable over Lewallen in view of Song. To establish a prima facie case of obviousness, the prior art reference (or references when combined) must teach or suggest each and every claim limitation.

Independent claim 1 as proposed to be amended recites, in relevant part:

updating, by the platform-independent data superstructure, information in a first segment of the platform-independent data superstructure associated with the at least one application event, responsive to receiving the at least one application event and independent of an update to a second segment in the platform-independent data superstructure; and

updating, in accordance with the superstructure segment update, the application state of the executing application on the device.

Independent claims 21, 38 and 91 recite similar limitations. As discussed in the Examiner's Interview of June 23, 2009, the cited references fail to disclose, teach or suggest all of the limitations of the pending claims because the cited references fail to disclose, teach or suggest updating the state of an executing application, in accordance with an update to a segment of a platform-independent data superstructure, the platform-independent data superstructure defining the appearance and behavior of an application and storing an application state, program code and internal logic of the application.

As discussed, Lewallen fails to teach or suggest updating an application state of an executing application, in accordance with an update to a segment of a platform-independent data superstructure as recited by the pending claims. Lewallen discloses a system for generating a user interface for a program by translating a standard application programming interface (API) into a user interface API supported by a particular platform in which the application will execute. *See* Lewallen, Abstract and at col. 5, lines 56-59. A system for translating an API of one type into an API of a second type does not teach a superstructure storing the program code of an executing application and used in updating an application state of the application during execution of the application. Nor does Lewallen suggest such a feature. One of ordinary skill in the art focused on improving a user interface for an existing Java application would not be motivated to modify Lewallen to provide for updating of an application state of an executing application based upon updates to a superstructure storing application state, program code, and internal logic of the application because a new method for creating an application and updating application state during execution of the application would not provide a means for improving the user interface of an existing application.

Song also fails to teach or suggest updating, in accordance with a superstructure segment update, the application state of the executing application on the device. In Song, a capture component generates a snapshot including session parameters representative of the current active state of a browser session. *See* Song at 6, [0064]. In order for an application to later access data stored by the snapshot, a restoration process occurs that involves conversion of the browser snapshot back into a browser state of an active session. *See* Song at 7, [0081]. Merely creating a

new snapshot in the Song system would not result in a modification of an application state of an executing application – the snapshot must be converted during a restoration process before the browser can access the data and pass the snapshot data to the executing application. In contrast, in the claimed invention, it is sufficient to update the platform-independent superstructure in order to update an application state of an executing application. Because an application in Song is a conventionally-coded application that cannot directly access the snapshot data or modify its execution state in response to an incremental update to a snapshot, Song fails to teach a structure that both contains an application in its entirety and is used, without conversion, to modify an application state of an executing application.

Nor does Song suggest such a feature. Song relies upon an existing execution environment and on the use of conventionally-coded applications in that environment and provides a system in which snapshot data is isolated from application execution. One of ordinary skill in the art would not be motivated to modify a system for data migration for existing applications to provide a system involving developing new applications whose execution is furthered by modifications to superstructures with which the applications are integrated because this would require re-writing applications in order to integrate the program code with the superstructure data and would not provide support for existing, conventionally-coded applications without modifying the applications.

Therefore, neither Lewallen nor Song teach or suggest modifying, in accordance with the superstructure segment update, the application state of the executing application on the device. Even if one of ordinary skill in the art focused on improving a user interface for an existing Java application modified the Lewallen system – which transforms APIs of one type into APIs of another within an existing program prior to execution of the application – to incorporate the approach in Song of creating an intermediate snapshot for use in recreating a browser state, the combination would fail to teach or suggest the updating, in accordance with the superstructure segment update, of the application state of an executing application.

Accordingly, Applicants submit that neither Lewallen nor Song, alone or in combination, disclose, teach or suggest each and every limitation of claims 1, 3-48, 50, 52, 70-73, 75-76, 79-

87, 89, and 91-108. Applicants respectfully request that the Examiner reconsider and withdraw the rejection of claims 1, 3-48, 50, 52, 70-73, 75-76, 79-87, 89, and 91-108.

Rejection of Claims 77-78 Under 35 U.S.C. §103(a)

The Examiner rejected claims 77-78 under 35 U.S.C. §103(a) as unpatentable over Lewallen, in view of Song, and further in view of Snyder. To establish a *prima facie* case of obviousness, the prior art reference (or references when combined) must teach or suggest each and every claim limitation.

The arguments made above with respect to the failure of Lewallen and Song to teach or suggest the each and every limitation of the pending claims, apply with equal force here and are reiterated as if set forth in full. Snyder also fails to teach updating, in accordance with the superstructure segment update, the application state of the executing application on the device. Snyder describes a system for selecting and displaying navigational information to provide more efficient and intuitive methods for planning and programming navigational paths. *See* Snyder, Abstract and col. 2, lines 55-60. This system for displaying navigational information does not teach a superstructure storing application state, program code, and internal logic of an application, or updating, in accordance with a superstructure segment update, the application state of the executing application on the device. Nor does Snyder suggest a superstructure storing application state, program code, or internal logic of an application or modifying, in accordance with the superstructure segment update, the application state of the executing application on the device. One of ordinary skill in the art would not be motivated to modify a system for intuitive display of navigational information to update the state of an executing application because new methods for the creation and execution of applications would not improve the display of navigational information for existing methods of planning navigational paths.

Accordingly, Applicants submit that neither Lewallen nor Song nor Snyder, alone or in combination, disclose, teach or suggest each and every limitation of claims 77-78. Applicants respectfully request that the Examiner reconsider and withdraw the rejection of claims 77-78.

CONCLUSION

In view of the above, each of the presently pending claims in this application is believed to be in immediate condition for allowance. Accordingly, the Examiner is respectfully requested to pass this application to issue.

Please charge any fees associated with this filing, or apply any credits, to our Deposit Account No. 03-1721.

Respectfully submitted,
CHOATE, HALL & STEWART LLP

Date: June 24, 2009

/John D. Lanza/
John D. Lanza
Registration No. 40,060

Patent Group
CHOATE, HALL & STEWART LLP
Two International Place
Boston, MA 02110
Tel: (617) 248-5000
Fax: (617) 502-5002